New Paradigm in Speech Recognition: Deep Neural Networks

Dominique Fohr, Odile Mella, Irina Illina

New Paradigm in Speech Recognition: Deep Neural Networks

Dominique Fohr, Odile Mella and Irina Illina

***Abstract*—This paper addresses the topic of deep neural networks (DNN). Recently, DNN has become a flagship in the fields of artificial intelligence. Deep learning has surpassed state-of-the-art results in many domains: image recognition, speech recognition, language modelling, parsing, information retrieval, speech synthesis, translation, autonomous cars, gaming, etc. DNN have the ability to discover and learn complex structure of very large data sets. Moreover, DNN have a great capability of generalization. More specifically, speech recognition with DNN is the topic of our work in this paper. We present an overview of different architectures and training procedures for DNN-based models. In the framework of transcription of broadcast news, our DNN-based system decreases the word error rate dramatically compared to a classical system.**

***Index Terms*—speech recognition, deep neural network, acoustic modeling**

I. Introduction

More and more information appear on Internet each day. And more and more information is asked by users. This information can be textual, audio or video and represents multimedia information. About 300 hours of multimedia is uploaded per minute [1]. It becomes difficult for companies to view, analyze, and mine the huge amount of multimedia data on the Web. In these multimedia sources, audio data represents a very important part. *Spoken content retrieval* consists in "machine listening" of data and extraction of information. Some search engines like Google, Yahoo, etc. perform the information extraction from text data very successfully and give a response very quickly. For example, if the user wants to get information about "Obama", the list of several textual documents will be given by Google in a few seconds of search. In contrast, information retrieval from audio documents is much more difficult and consists of "machine listening" of the audio data and detecting instants at which the keywords of the query occur in the audio documents. For example, to find all audio documents speaking about "Obama".

Not only individual users, but also a wide range of companies and organizations are interested by these types of applications. Many business companies are interested to know what is said about them and about their competitors on broadcast news or on TV. In the same way, a powerful indexing system of audio data would benefit archives. Well organized historical archives can be rich in term of cultural value and can be used by researchers or general public.

Classical approach for spoken content retrieval from audio documents is speech recognition followed by text retrieval [2]. In this approach, the audio document is transcribed automatically using a speech recognition engine and after this the transcribed text is used for the information retrieval or opinion mining. The speech recognition step is crucial, because errors occurring during this step will propagate in the following step.

In this article, we will present the new paradigm used for speech recognition: *Deep Neural Networks* (DNN). This new methodology for automatic learning from examples achieves better accuracy compared to classical methods.
In section II, we briefly present automatic speech recognition. Section III gives an introduction to deep neural networks. Our speech recognition system and an experimental evaluation are described in section IV.

II. Automatic Speech Recognition

An automatic speech recognition system requires three main sources of knowledge: an *acoustic model*, a *phonetic lexicon* and a *language model* [3]. Acoustic model characterizes the sounds of the language, mainly the phonemes and extra sounds (pauses, breathing, background noise, etc.). The phonetic lexicon contains the words that can be recognized by the system with their possible pronunciations. Language model provides knowledge about the word sequences that can be uttered. In the state-of-the-art approaches, statistical acoustic and language models, and to some extent lexicons, are estimated using huge audio and text corpora.

Automatic speech recognition consists in determining the best sequence of words ($\widehat{W}$) that maximize the likelihood:

$$\widehat{W} = \text{argmax}_{W} P(X|W)P(W) \qquad (1)$$

where *P(X|W)*, known as *acoustic probability*, is the probability of the audio signal (*X*) given the word sequence *W*. This probability is computed using acoustic model. *P(W)*, known as *language probability*, is the probability *a priori* of the word sequence, computed using the language model.

This work was funded by the *ContNomina* project supported by the French National Research Agency (ANR) under contract ANR-12-BS02-0009.

All authors are with the Université de Lorraine, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France, Inria, Villers-lès-Nancy, F-54600, France, CNRS, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France (e-mail: fohr@ loria.fr, illina@loria.fr, mella@loria.fr).

A. Acoustic modeling

Acoustic modeling is mainly based on *Hidden Markov Model* (HMM). An HMM is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved (*hidden*) states [4].

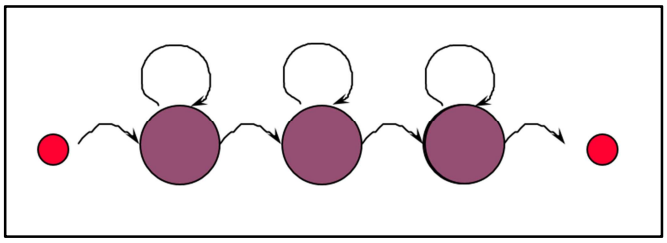

Fig. 1. HMM with 3 states, left-to-right topology and self-loops, commonly used in speech recognition.

HMM is a finite state automaton with *N* states, composed of three components: $\{A, B, \Pi\}$. A is the transition probability matrix (a_{ij} is the transition probability from the state *i* to the state *j*). Π is the prior probability vector (π_i the prior probability of state *i)*, and B is the emission probability vector ($b_j(x)$ *is* the probability of emission of observation *x* being in state *j)*.

In speech recognition, the main advantage of using HMM is its ability to take into account the dynamic aspects of the speech. When a person speaks quickly or slowly, the model can correctly recognize the speech thanks to the self-loop on the states.

To model the sounds of a language (phones), a three-state HMM is commonly chosen (cf. Fig. 1). These states capture the beginning, central and ending parts of a phone. In order to capture the coarticulation effects, *triphone models* (a phone in a specific context of previous and following phones) are preferred to context-independent phone models.

Until 2012, emission probabilities were represented by a mixture of multivariate Gaussian probability distribution functions modeled as:

$$b_j(x) = \sum_{m=1}^{M} c_{jm}\, \mathcal{N}(x; \mu_{jm}, \Sigma_{jm}) \qquad (2)$$

The parameters of Gaussian distributions are estimated using the *Baum-Welch* algorithm.

A tutorial on HMM can be found in [4]. These models were successful and achieved best results until 2012.

B. Language modeling

Historically, the most common approach for language modeling is based on statistical *n*-gram model. An *n*-gram model gives the probability of a word w_i given the *n-1* previous words:

$$P(w_i|w_1, \ldots, w_{i-1}) = P(w_i|w_{i-(n-1)}, w_{i-(n-2)}, \ldots, w_{i-1})$$

These probabilities are estimated on a huge text corpus. To avoid a zero probability for unseen word sequences, smoothing methods are applied, the best known smoothing method being proposed by Kneiser-Ney [5].

C. Search for the best sentence

The optimal computation of the sentence to recognize is not tractable because the search space is too large. Therefore, heuristics are applied to find a good solution. The usual way is to perform the recognition in two steps:

- The aim of this first step is to remove words that have a low probability to belong to the sentence to recognize. A word lattice is constructed using beam search. This word lattice contains best word hypotheses. Each hypothesis consist of words, their acoustic probabilities, language model probabilities and time boundaries of the words.
- The second step consists in browsing the lattice using additional knowledge to generate the best hypothesis.

Usually, the performance of automatic speech recognition is evaluated in terms of *Word Error Rate* (WER), i.e. the number of errors (insertions, deletion and substitutions) divided by the number of words in the test corpus.

III. Deep Neural Networks

In 2012, an image recognition system based on *Deep Neural Networks* (DNN) won the *Image net Large Scale Visual Recognition Challenge* (ILSVCR) [6]. Then, DNN were successfully introduced in different domains to solve a wide range of problems: speech recognition [7], speech understanding, parsing, translation [8], autonomous cars [9], etc.[10]. Now, DNN are very popular in different domains because they allow to achieve a high level of abstraction of large data sets using a deep graph with linear and non-linear transformations. DNN can be viewed as universal approximators. DNN obtained spectacular results and now their training is possible thanks to the use of GPGPU (*General-Purpose Computing on Graphics Processing Units*).

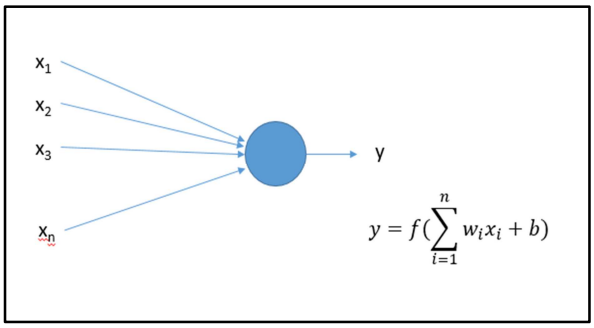

Fig. 2. Example of one neuron and its connections.

A. Introduction

Deep Neural Networks are composed of *neurons* that are interconnected. The neurons are organized into layers. The first layer is the input layer, corresponding to the data features. The last layer is the output layer, which provides the output probabilities of classes or labels (classification task).

The output *y* of the neuron is computed as the non-linear weighted sum of its input. The neuron input x_i can be either the input data if the neuron belongs to the first layer, or the

output of another neuron. An example of a single neuron and its connections is given in Figure 2.

A DNN is defined by three types of parameters [11]:

- The interconnection pattern between the different layers of neurons;
- The training process for updating the weights w_i of the interconnections;
- The activation function f that converts a neuron's weighted input to its output activation (cf. equation in Fig. 2).

The widely used activation function is the *non-linear weighted sum*. Using only linear functions, neural networks can separate only linearly separable classes. Therefore, non-linear activation functions are essential for real data. Figure 3 shows some classical non-linear functions as sigmoid, hyperbolic tangent (*tanh*), *RELU* (*Rectified Linear Units*), and *maxout*.

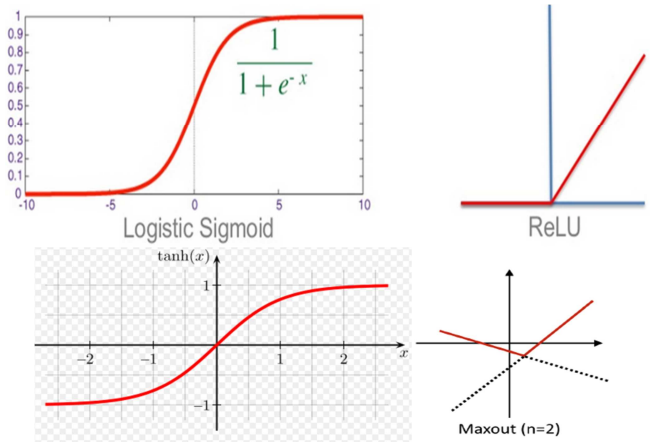

Fig. 3. Sigmoid, *RELU*, tangent hyperbolic and maxout non-linear functions

B. Training

The goal of the training is to reduce the error between the outputs computed on the training data and the target values. This supervised training consists in estimating the weights w_i. of all neurons of all layers. Until now, there is no learning process that converges to a global optimum. The classical learning algorithm is based on stochastic gradient descent and only a local optimum can be achieved.

At each *epoch* (one *epoch* consists of one training cycle on the whole training set), the cost function E (difference between target output and computed output) is computed and the weights ω are adjusted using:

$$\Delta\omega = -\eta \frac{dE}{d\omega}$$

η is called the learning rate. In general, the learning rate is decreased during the training process [12][13].

Theoretically, the gradient should be computed using the whole training corpus. However, the convergence is very slow because the weights are updated only once per *epoch*. One solution of this problem is to use *Stochastic Gradient Descent* (SGD). It consists in computing the gradient on a small set of training samples (called *mini-batch*) and in updating the weights after each mini-batch. This speeds up the training process.

During the training, it may happen that the network learns features or correlations that are specific to the training data rather than generalize the training data to be applicable to the test data. This phenomenon is called *overfitting*. One solution is to use a development set that should be as close as possible to the test data. On this development set, recognition error is calculated at each epoch of the training. When the error begins to increase, the training is stopped. This process is called *early stopping*. Another solution to avoid overfitting consists in using *regularization*. It consists in inserting a constraint to the error function to restrict the search space of weights. For instance, the sum of the absolute values of the weights can be added to the error function [14].

One more solution to avoid overfitting is *dropout* [15]. The idea is to "remove" randomly some neurons during the training. This prevents neurons from co-adapting too much and performs model averaging.

C. Different DNN architectures

There are different types of DNN regarding the architecture [16]:

- *MultiLayer Perceptron* (MLP): each neuron of a layer is connected with all neurons of the previous layer (feedforward and unidirectional).
- *Recurrent Neural Network* (RNN): when it models a sequence of inputs (time sequence), the network can use information computed at previous time (*t-1*) while computing output for time *t*. Fig. 4 shows an example of a RNN for language modeling: the hidden layer *h(t-1)* computed for the word *t-1* is used as input for processing the word *t* [17].

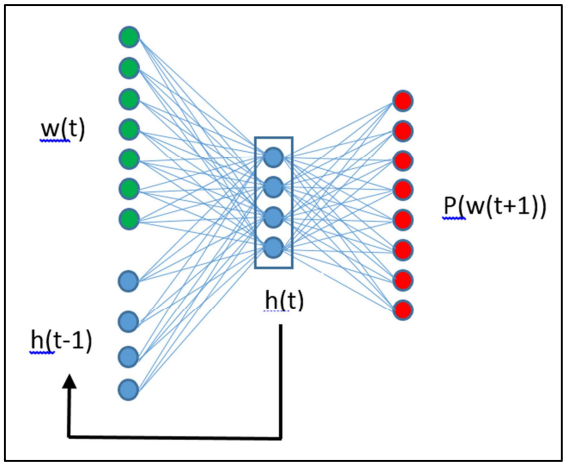

Fig. 4. Example of a RNN.

- *Long Short-Term Memory* (LSTM) is a special type of RNN. The problem with RNN is the fact that the gradient is vanishing, and the memory of past events decreases. Sepp Hochreiter and Jürgen

Schmidhuber [18] have proposed a new recurrent model that has the capacity to recall past events. They introduced two concepts: *memory cell* and *gates*. These gates determine when the input is significant enough to remember or forget the value, and when it outputs a value. Fig. 5 displays the structure of an LSTM.

- *Convolutional Neural Network* (CNN) is a special case of Feedforward Neural Network. The layer consists of filters (cf. Fig. 6). The parameters of these filters are learned. One advantage of this kind of architecture is the sharing of parameters, so there are fewer parameters to estimate. In the case of image recognition, each filter detects a simple feature (like a vertical line, a contour line, etc.). In deeper layer, the features are more complex (cf. Fig. 7). Frequently, a *pooling* layer is used. This layer allows a non-linear *downsampling*: max pooling (cf. Fig. 8) computes maximum values on sub-region. The idea is to reduce the size of the data for the following layers. An example of state-of-the-art acoustic model using CNN is given in Fig. 9.

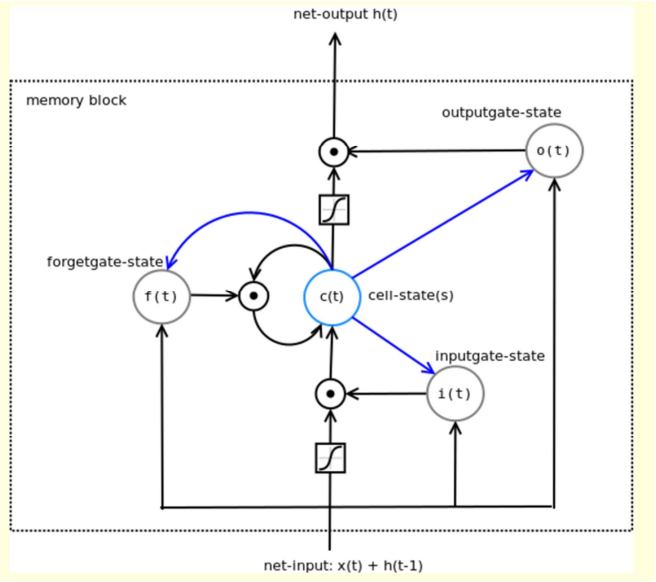

Fig. 5. Example of LSTM with three gates: input gate, forget gate, output gate and a memory cell (from [19]).

The main advantage of RNN and LSTM is their ability to take into account temporal evolution of the input features. These models are widely used for natural language processing. Strong point of CNN is the translation invariance, i.e. the skill of discover structure patterns regardless the position. For acoustic modelling all these structures can be exploited.

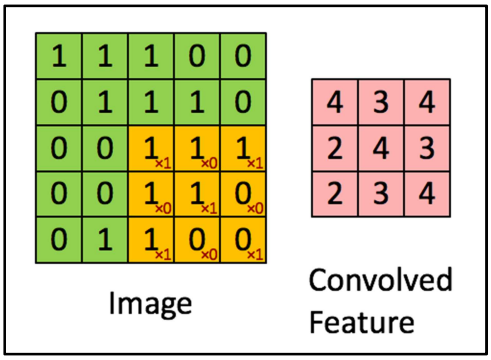

Fig. 6. Example of a convolution with a filter $\begin{bmatrix}1 & 0 & 1\\0 & 1 & 0\\1 & 0 & 1\end{bmatrix}$
Original image is in green, filter applied on bottom right of image is in orange and convolution result is in pink.

A difficult DNN issue is the choice of the hyperparameters: number of hidden layers, number of neurons per layer, choice of non-linear functions, choice of learning rate adaptation function. Often, some hyperparameters are adjusted experimentally (trial and error), because they depend on the task, the size of the database and data sparsity.

D. DNN-based acoustic model

As said previously, for acoustic modeling, HMM with 3 left-to-right states are used to model each phone or contextual phone (triphone). Typically, there are several thousand of HMM states in a speech recognition system.

In DNN-based acoustic model, contextual phone HMMs are keeped but all the Gaussian mixtures of the HMM states (equation 2) are replaced by DNN. Therefore, DNN-based acoustic model computes the observation probability $b_j(x)$ of each HMM phone state given the acoustic signal using DNN networks [21]. The input of the DNN will be the acoustic parameters at time *t*. The DNN outputs correspond to all HMM states, one output neuron for one HMM state.

In order to take into account contextual effects, the acoustic vectors from a time window centered on time t (for instance from time t-5 to t+5) are put together.

To train the DNN acoustic model, the alignment of the training data is necessary: for each frame, the corresponding HMM state that generated this frame should be known. This alignment of the training data is performed using a classical GMM-HMM model.

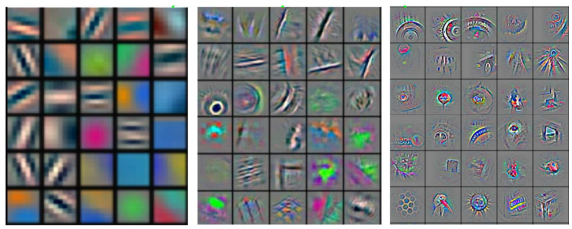

Fig. 7. Feature visualization of convolutional network trained on ImageNet from Zeiler and Fergus [20].

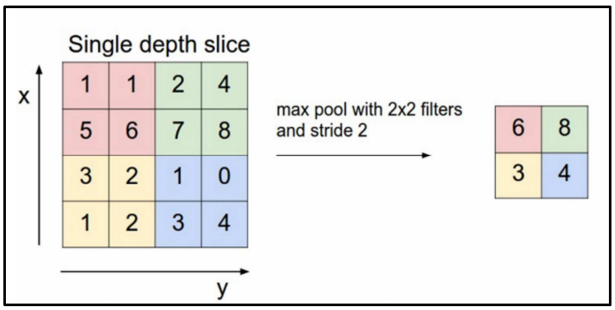

Fig. 8. Max pooling with a 2x2 filter (from www.wildml.com)

E. Language model using DNN

A drawback of classical N-gram language models (LM) is their weak ability of generalization: if a sequence of words was not observed during training, N-gram model will give poor probability estimation. To address this issue, one solution is to move to a continuous space representation. Neural networks are efficient for carrying out such a projection. To take into account the temporal structure of language (word sequences), RNN have been largely studied. The best NN-based language models use LSTM and RNN [23][24].

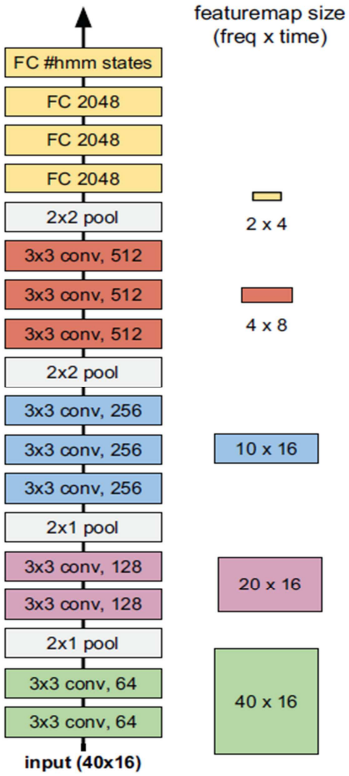

Fig. 9. The very deep convolutional system proposed by IBM for acoustic modeling: 10 CNN, 4 pooling, 3 full connected (FC MLP) (from [22]).

IV. KATS (KALDI BASED TRANSCRIPTION SYSTEM)

In this section we present the KATS speech recognition system developed in our speech group. This system is built using *Kaldi* speech recognition toolkit, freely available under the Apache License. Our KATS system can use GMM-based and DNN-based acoustic models.

A. Corpus

The training and test data were extracted from the *radio broadcast news* corpus created in the framework of the ESTER project [25]. This corpus contains 300 hours of manually transcribed shows from French-speaking radio stations (*France Inter*, *Radio France International* and *TVME Morocco*). Around 250 h were recorded in studio and 50h on telephone. 11 shows corresponding to 4 hours of speech (42000 words) were used for evaluation.

B. Segmentation

The first step of our KATS system consists in segmentation and diarization. This module splits and classifies the audio signal into homogeneous segments: non-speech segments (music and silence), telephone speech and studio speech. For this, we used the toolkit developed by LIUM [27]. We processed separately telephone speech and studio speech in order to estimate two sets of acoustic models; studio models and telephone models.

C. Parametrization

The speech signal is sampled at 16 kHz. For analysis, 25 ms frames are used, with a frame shift of 10 ms. 13 MFCC were calculated for each frame completed by the 13 delta and 13 delta-delta coefficients leading to a 39-dimension observation vector. In all experiments presented in this paper, we used MCR (*Mean Cepstral Removal*).

D. Acoustic models

In order to compare GMM-HMM and DNN-HMM acoustic models, we used the same HMM models with 4048 senones. The only difference is the computation of the emission probability ($b_j(x)$ of equation 2): for GMM-HMM it is a mixture of Gaussians, for DNN-HMM, it is a deep neural network. Language model and lexicon stay the same. For GMM-HMM acoustic models, we used 100k Gaussians. For DNN, the input of the network is the concatenation of 11 frames (from *t-5* to *t+5*) of 40 parameters. The network is a MLP with 6 hidden layers of 2048 neurons per layer (cf. Fig. 10). The output layer has 4048 neurons (corresponding to 4048 senones). The total number of parameters in DNN-HMM is about 30 millions.

E. Language models and lexicon

Language models were trained of huge text corpora: newspaper corpus (*Le Monde*, *L'Humanité*), news wire (*Gigaword*), manual transcriptions of training corpus and web data. The total size was 1.8 billion words. The *n*-gram language model is a linear combination of LM models trained on each text corpus. In all experiments presented in this paper, only a 2-gram model is used with 40 million bigrams and a

lexicon containing 96k words and 200k pronunciations.

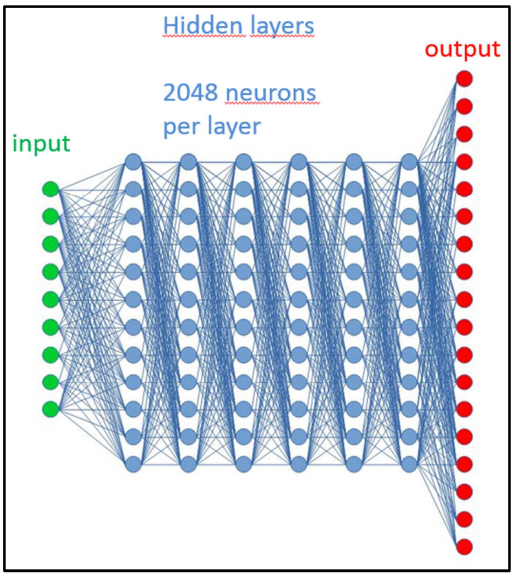

Fig. 10. Architecture of the DNN used in KATS system.

F. Recognition results

Recognition results in terms of word error rate for the 11 shows are presented in Table 1. The confidence interval of these results is about +/- 0.4 %. Two systems are compared. These systems use the same lexicon and the same language models but differ by their acoustic models: GMM-HMM and DNN-HMM, so, the comparison is fair. For all shows, the DNN-based system outperforms the GMM-based system. The WER difference is 5.3% absolute, and 24% relative. The improvement is statistically significant. The large difference in performance between the two systems suggests that DNN-based acoustic models achieves better classification and has generalization ability.

Shows	# words	GMM-HMM	DNN-HMM
20070707_rfi (France)	5473	23.6	16.5
20070710_rfi (France)	3020	22.7	17.4
20070710_france_inter	3891	16.7	12.1
20070711_france_inter	3745	19.3	14.4
20070712_france_inter	3749	23.6	16.6
20070715_tvme (Morocco)	2663	32.5	26.5
20070716_france_inter	3757	20.7	17.0
20070716_tvme (Morocco)	2453	22.8	17.0
20070717_tvme (Morocco)	2646	25.1	20.1
20070718_tvme (Morocco)	2466	20.2	15.8
20070723_france_inter	8045	22.4	17.4
Average	41908	22.4	**17.1**

Table 1. Word Error Rate (%) for the 11 shows obtained using the GMM-HMM and DNN-HMM KATS systems.

V. Conclusion

From 2012, deep learning has shown excellent results in many domains: image recognition, speech recognition, language modelling, parsing, information retrieval, speech synthesis, translation, autonomous cars, gaming, etc. In this article, we presented deep neural networks for speech recognition: different architectures and training procedures for acoustic and language models are visited. Using our speech recognition system, we compared GMM and DNN acoustic models. In the framework of broadcast news transcription, we shown that the DNN-HMM acoustic model decreases the word error rate dramatically compared to classical GMM-HMM acoustic model (24% relative significant improvement).

The DNN technology is now mature to be integrated into products. Nowadays, main commercial recognition systems (Microsoft *Cortana*, Apple *Siri*, Google *Now* and Amazon *Alexa*) are based on DNNs.

Acknowledgment

This work was funded by the *ContNomina* project supported by the French National Research Agency (ANR) under contract ANR-12-BS02-0009.

References

[1] L.-S. Lee, H.-Y. Lee (2016). Spoken Content Retrieval - Beyond Cascading Speech Recognition with Text Retrieval. Tutorial of Interspeech.

[2] M. Larson and G. J. F. Jones (2012). Spoken Content Retrieval: A Survey of Techniques and Technologies. *Foundations and Trends in Information Retrieval, vol.* 5, *no*. 4-5.

[3] L. Deng, X. Li (2013). Machine Learning Paradigms for Speech Recognition, IEEE Transactions on Audio, Speech and Language Processing, vol. 2, n. 5.

[4] L. Rabiner. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE 77, no. 2, p. 257-86.

[5] R. Kneser, H. Ney, (1995). Improved Backing-off for m-gram Language Modeling. IEEE International Conference on Acoustics, Speech and Signal Processing, Detroit, MI, volume 1, pp. 181–184.

[6] A. Krizhevsky I. Sutskever G. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks, NIPS.

[7] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, G. Zweig (2016). Achieving Human Parity in Conversational Speech Recognition. arXiv:1610.05256

[8] W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, arXiv:1609.08144.

[9] M. Bojarski, D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, K. Zieba (2016). End to End Learning for Self-Driving Cars, arXiv:1604.07316.

[10] L. Deng (2014). A Tutorial Survey of Architectures, Algorithms and Applications for Deep Learning, APSIPA Transactions on Signal and Information Processing.

[11] https://en.wikipedia.org/wiki/Artificial_neural_network

[12] D. Kingma, J. Ba, (2015). Adam: a Method for Stochastic Optimization. International Conference on Learning Representations.

[13] J. Duchi, , E. Hazan, , Y. Singer (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, 12, 2121–2159.

[14] I. Goodfellow, Y. Bengio, A. Courville, (2016). Deep Learning, Book MIT Press.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting 15(Jun):1929−1958.
[16] Y. LeCun, Y. Bengio, G. Hinton (2015). Deep Learning, Nature, vol. 521, May 2015.
[17] T. Mikolov (2012). Statistical Language Models based on Neural Networks. PhD thesis, Brno University of Technology.
[18] S. Hochreiter, J. Schmidhuber (1997). Long Short-Term Memory, Neural Computation. 9 (8): 1735–1780.
[19] http://christianherta.de/lehre/dataScience/machineLearning/neuralNetworks/LSTM.php
[20] M. Zeiler, R. Fergus (2013). Visualizing and Understanding Convolutional Networks, arXiv:1311.2901.
[21] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, vol. 82.
[22] G. Saon, T. Sercu, S. Rennie, H.-K. Kuo (2016). The IBM 2016 English Conversational Telephone Speech Recognition System, Interspeech.
[23] M. Sundermeyer, H. Ney, R. Schlüter (2015). From Feedforward to Recurrent LSTM Neural Networks for Language Modeling, *IEEE Transactions on Audio, Speech and Language Processing, vol.. 23, no. 3, March 2015.*
[24] K. Irie, Z. Tüske, T. Alkhouli, R. Schlüter, H. Ney (2016). LSTM, GRU, Highway and a Bit of Attention: An Empirical Overview for Language Modeling in Speech Recognition, Interspeech.
[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlıcek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, K. Vesely (2011). The Kaldi Speech Recognition Toolkit, ASRU.
[26] S. Galliano, G. Gravier, L. Chaubard (2009). The ESTER 2 Evaluation Campaign for the Rich Transcription of French Radio Broadcasts. Interspeech.
[27] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, S. Meignier (2013). An Open-source State-of-the-art Toolbox for Broadcast News Diarization,“ Interspeech, Lyon (France), 25-29.